



TUGAS AKHIR – TF 145565

**RANCANG BANGUN SISTEM PENGENDALIAN
LEVEL TANGKI PENAMPUNG PADA SIMULATOR
ALAT PENUKAR PANAS**

FARADHIBA ALIFIYAH SAFITRI
NRP 10 51 15 00000 003

Dosen Pembimbing I
Dr. Ir. Totok Suhartanto, DEA.
NIP. 19650309 199002 1 001

Dosen Pembimbing II
Dwi Oktavianto Wahyu Nugroho, ST., M.T.
NPP. 19832017 1105 4

PROGRAM STUDI D3 TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNOLOGI INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018



TUGAS AKHIR – TF 145565

**RANCANG BANGUN SISTEM PENGENDALIAN
LEVEL TANGKI PENAMPUNG PADA SIMULATOR
ALAT PENUKAR PANAS**

FARADHIBA ALIFIYAH SAFITRI
NRP 10 51 15 00000 003

Dosen Pembimbing I
Dr. Ir. Totok Soehartanto, DEA.
NIP. 19650309 199002 1 001

Dosen Pembimbing II
Dwi Oktavianto Wahyu Nugroho, ST., M.T.
NPP. 19832017 1105 4

PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNOLOGI INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018



FINAL PROJECT – TF 145565

***DESIGN OF LEVEL CONTROL SYSTEM IN STORAGE
TANK MINI PLANT HEAT EXCHANGER***

FARADHIBA ALIFIYAH SAFITRI
NRP 10 51 15 00000 003

Supervisor I
Dr. Ir. Totok Soehartanto, DEA.
NIP. 19650309 199002 1 001

Supervisor II
Dwi Oktavianto Wahyu Nugroho, ST., M.T.
NPP. 19832017 1105 4

***DIII INSTRUMENTATION ENGINEERING
DEPARTMENT OF INSTRUMENTATION ENGINEERING
FACULTY OF VOCATION
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018***

LEMBAR PENGESAHAN
RANCANG BANGUN SISTEM PENGENDALIAN
LEVEL TANGKI PENAMPUNG PADA SIMULATOR
ALAT PENUKAR PANAS

TUGAS AKHIR

Oleh :

Faradhiba Alifiyah Safitri
NRP 10 51 15 00000 003

Surabaya, 30 Juli 2018
Mengetahui dan Menyetujui,

Dosen Pembimbing I,



Dr. Ir. Totok Sochartanto, DEA.
NIP. 19650309 199002 1 001

Dosen Pembimbing II,



Dwi Oktavianto Wahyu N. S.T, M.T
NPP. 19832017 1105 4



Kepala Departemen
Teknik Instrumentasi,

Dr. Ir. Purwadi Agus Darwito, M.Sc
NIP. 19620822 198803 1 001


**RANCANG BANGUN SISTEM PENGENDALIAN
LEVEL TANGKI PENAMPUNG PADA SIMULATOR
ALAT PENUKAR PANAS**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
Pada
Program Studi DIII Teknologi Instrumentasi
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

**Oleh :
Faradhiba Alifiyah Safitri
NRP 10 51 15 00000 003**

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. Ir. Totok Soehartanto, DEA..... (Dosen Pembimbing I)
 2. Dwi Oktavianto Wahyu N. S.T, M.T..... (Dosen Pembimbing II)
 3. Ahmad Fauzan Adziimaa S.T, M.Sc (Dosen Penguji)
- 

SURABAYA
JULI 2018

RANCANG BANGUN SISTEM PENGENDALIAN LEVEL TANGKI PENAMPUNGAN PADA SIMULATOR ALAT PENUKAR PANAS

Nama : Faradhiba Alifiyah Safitri
NRP : 10511500000003
Jurusan : Teknik Instrumentasi
Pembimbing 1 : Dr. Ir. Totok Soehartanto, DEA
Pembimbing 2 : Dwi Oktavianto Wahyu Nugroho, S.T., M.T.

ABSTRAK

Pada tugas akhir ini telah dibuat simulator alat penukar panas yang dilengkapi dengan tangki penampungan. Tangki penampungan ini berfungsi sebagai tempat menampung air yang keluar dari *cooling tower* agar dapat disirkulasi kembali. Oleh karena itu, perlu dilakukan rancang bangun sistem pengendalian level yang berfungsi untuk menjaga kestabilan level pada tangki penampungan agar ketersediaan air selama proses tetap terjaga. Pada sistem pengendalian level ini menggunakan sensor level berbasis kapasitansi. yang akan mendeteksi ketinggian air. Hasil pengukuran level ini akan diolah pada kontroler yaitu Atmega16 untuk menghasilkan *output* yang dapat mengindikasikan kondisi level tangki penampungan, sehingga ketika level pada tangki penampungan kurang maka aktuator yaitu *motor operated valve* akan membuka air tambahan dari tangki *makeup water*. Pada sistem pengendalian ini menggunakan mode kontrol proporsional dengan nilai K_p 6 dan set point 30 cm. Metode yang digunakan pada sistem ini adalah *trial and error*. Hasil dari pengujian sistem didapatkan waktu *rise time* (T_r) pada 720 detik, *peak time* pada 810 detik dan waktu *steady* pada 900 detik..

Kata kunci : *Simulator alat penukar panas, makeup water, motor operated valve, rise time, , peak time, steady.*

DESIGN OF LEVEL CONTROL SYSTEM IN STORAGE TANK MINI PLANT HEAT EXCHANGER

Name : Faradhiba Alifiyah Safitri
NRP : 105115000000003
Departement : Instrumentation Engineering
Supervisor I : Dr. Ir. Totok Soehartanto, DEA
Supervisor II : Dwi Oktavianto Wahyu Nugroho, S.T., M.T.

ABSTRACT

In this final project has been made mini plant heat exchanger which is equipped with storage tank. This storage tank serves as a place to store water coming out of the cooling tower in order to be recirculated again. Therefore, it is necessary to design a level control system that serves to maintain the level of water storage tanks to keep the process maintained. At this level control system uses level sensor based on capacitance. which will detect the height of the water. The results of this level measurement will be processed on the controller that is Atmega16 to produce output that can indicate the condition level of the storage tank, so that when level on the storage tank is less then set point, the actuator is motor operated valve will open additional water from the makeup water tank.. In this control system using the control mode proportional with the value of k_p 6 and set point 30 cm. The method of this control system is trial and error. The results of the level control system test the time rise time (T_r) at 720 seconds, peak time at 810 seconds and stable time at 900 seconds

Keywords: *miniplant heat exchanger, makeup water, motor operated valve, rise time,, peak time, steady.*

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas berkat, rahmat dan kebesaran-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul “RANCANG BANGUN SISTEM PENGENDALIAN LEVEL TANGKI PENAMPUNGAN PADA SIMULATOR ALAT PENUKAR PANAS” tepat pada waktunya.

Selama menyelesaikan tugas akhir ini penulis telah banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. Ir.Purwadi Agus Darwito,M.Sc selaku Kepala Departmen D3 Teknik Instrumentasi ITS.
2. Bapak Dr. Ir. Totok Soehartanto, DEA, selaku dosen pembimbing 1 yang telah meluangkan waktu, arahan, saran , semangat dan motivasi yang membantu selama proses pengerjaan tugas akhir ini.
3. Bapak Dwi Oktavianto Wahyu Nugroho, S.T.,M.T, selaku dosen pembimbing 2 yang telah meluangkan waktu, arahan dan saran yang membantu selama proses pengerjaan tugas akhir ini.
4. Bapak Detak Yan Pratama, S.T, M.Sc, selaku dosen wali.
5. Orang tua dan keluarga tercinta yang selalu memberi dukungan dan semangat agar tidak jatuh dalam keterpurukan.
6. Terimakasih kepada mas dayat, mas khafid, mas berrel, mas cahyo yang telah membantu dan memberi saran dalam pengerjaan tugas akhir ini.
7. Terimakasih kepada Jombierumpita, mbak lady, vebby, nyaq, cici, shaski, yang selalu berusaha mendinginkan suasana.
8. Teman-teman terkasih Teknik Intrumentasi 2015 yang selalu senantiasa memberikan semangat dan semua pihak yang telah membantu terselesaikannya Tugas Akhir ini.
9. Serta semua pihak yang tidak dapat penulis sebutkan satu persatu.

Penulis menyadari bahwa kesempurnaan hanya milik Allah SWT. Oleh sebab itu, penulis sangat berterimakasih atas segala masukan, kritik dan saran yang membangun dari pembaca agar laporan ini menjadi lebih baik untuk di kemudian hari. Demikian laporan ini penulis buat, semoga laporan ini dapat memberikan manfaat selain bagi penulis sendiri, dan bagi pembaca sekalian.

Surabaya, 24 Juli 2018

Penulis

Faradhiba Alifiyah Safitri
NRP. 10511500000003

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii

BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Lingkup Tugas Akhir	3

BAB II DASAR TEORI

2.1 Simulator Alat Penukar Panas	5
2.2 Tangki Penampungan	6
2.3 Sensor Kapasitif.....	8
2.4 <i>Stepper Motor Controlled Valve</i>	9
2.5 Mikrokontroler AVR ATmega16	11
2.6 CodeVision AVR.....	12
2.7 Sistem Pengendalian Level.....	15

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Diagram Alir.....	19
3.2 Perancangan Sistem.....	21
3.3 Pembuatan Sistem	25

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil Rancang Bangun Alat	29
4.2 Pengujian Sistem	34
4.3 Pembahasan	37

BAB V PENUTUP

5.1 Kesimpulan.....	39
5.2 Saran	39

DAFTAR PUSTAKA

LAMPIRAN A (LISTING PROGRAM)

LAMPIRAN B (DATASHEET ATMEGA 16)

LAMPIRAN C (MANUAL BOOK POMPA)

LAMPIRAN D (DOKUMENTASI)

DAFTAR GAMBAR

Gambar 2.1	P&ID Simulator Alat Penukar Panas.....	5
Gambar 2.2	BFD Simulator Alat Penukar Panas	5
Gambar 2.3	Desain Tangki Penampungan	7
Gambar 2.4	Konsep Dasar Kapasitansi.....	8
Gambar 2.5	<i>Stepper Motor Controlled Valve</i>	10
Gambar 2.6	Motor Stepper dengan Lilitan Unipolar.....	12
Gambar 2.7	Mikrokontroller AVR ATmega16.....	13
Gambar 2.8	Pin – pin AVR ATmega16	13
Gambar 2.9	Diagram Blok Sistem Pengendalian Level ...	16
Gambar 3.1	Diagram Alir Pembuatan Alat	19
Gambar 3.2	P&ID Simulator Alat Penukar Panas.....	20
Gambar 3.3	Perancangan Sistem Pengendalian Level pada Tangki Penampungan	20
Gambar 3.4	Diagram Blok Sistem Pengendalian Level ...	21
Gambar 3.5	Desain Tangki Penampungan	21
Gambar 3.6	Desain Tangki <i>water makeup</i>	22
Gambar 3.7	<i>Flowchart</i> Program.....	23
Gambar 3.8	Tampilan Listing Program <i>Software Code</i> Vision AVR	24
Gambar 3.9	Tampilan AVR Dude 5.2	24
Gambar 3.10	Tangki Penampungan	25
Gambar 3.11	Tangki <i>makeup water</i>	26
Gambar 3.12	Pompa	26
Gambar 3.13	<i>Motor Operating Valve</i>	27
Gambar 4.1	Plant Simulator Alat Penukar Panas	29
Gambar 4.2	Sistem Pengendalian Level.....	29
Gambar 4.3	Rangkaian <i>Motor Valve</i>	30
Gambar 4.4	grafik pembacaan sensor dan penggaris	29
Gambar 4.5	Grafik Respon Sistem Pada Set Point 30 cm dengan Nilai Kp 6.....	30
Gambar 4.6	Grafik Respon Bukaan Valve Terhadap Level pada Set Point 30 cm	30

DAFTAR TABEL

Tabel 2.1 Dielektrik <i>Constant</i> Bahan	8
Tabel 2.2 Fungsi khusus port B.....	14
Tabel 2.3 Fungsi khusus port C.....	14
Tabel 2.4 Fungsi khusus port D.....	15
Tabel 4.1 Hasil pengujian stepper	31
Tabel 4.2 pengambilan data naik.....	32
Tabel 4.3 Pengambilan Data Turun.....	32
Tabel 4.4 Rata – Rata Pembacaan	33
Tabel 4.5 Hasil Pengujian Sistem.....	34
Tabel 4.4 Rata – Rata Pembacaan	33

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pada simulator alat penukar panas yang dibuat kelompok tugas akhir ini, tangki penampung berfungsi sebagai tempat untuk menampung fluida berupa air sebelum di salurkan menuju tangki pemanas dan tangki pendingin dengan cara mengatur debit dari kedua tangki untuk diteruskan menuju *heat exchanger*. Air yang keluar dari *heat exchanger* didinginkan pada *cooling tower* agar suhunya kembali normal. Setelah suhu normal maka air dari *cooling tower* akan menuju tangki penampungan untuk disirkulasikan kembali.

Debit keluaran *cooling tower* yang masuk pada tangki penampungan nilainya lebih kecil daripada yang dipompa menuju tangki pemanas dan pendingin. Peranan pengendalian level pada dasarnya bertujuan agar proses berjalan sesuai dengan apa yang diinginkan karena, jika ketinggian fluida tidak dikendalikan maka proses akan terganggu. Seperti ketinggian fluida melebihi ketinggian yang diinginkan maka akan terjadi *overflow* atau fluida akan meluap sehingga mengganggu atau dapat merusak alat-alat lain dan jika ketinggian fluida kurang dari ketinggian yang diinginkan maka proses tidak akan bekerja.[1] Untuk itu, ketersediaan air pada tangki harus dijaga sesuai *set point*. agar ketinggiannya stabil dengan mengalirkan air dari *make up water tank*. dan menghindari adanya udara yang ikut terpompa dengan merancang sistem pengendalian level pada tangki penampungan. Selain itu, instrumen yang digunakan harus mampu menghasilkan sinyal elektrik agar dapat dimonitoring pada HMI (*Human Machine Interface*) dan hasil pengukurannya dapat ditampilkan pada LCU (*Local Control Unit*).

Dari tugas akhir ini, akan dihasilkan sistem pengendalian level pada tangki penampungan air dengan menggunakan sensor level yang akan mendeteksi ketinggian air dan mengirim sinyal ke *controller* untuk menggerakkan aktuator yang akan membuka aliran air pada pipa keluaran dari *make up water tank* jika level yang dideteksi tidak sesuai dengan *set point*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah dalam tugas akhir ini adalah :

1. Bagaimana merancang tangki penampung air dari *output cooling tower* berdasarkan hukum kesetimbangan *massa*?
2. Bagaimana merancang sistem pengendalian level pada tangki penampung air agar sesuai *set point* dan dapat diamati pada *control room*?

1.3 Tujuan

Adapun tujuan dari rancang bangun alat ini adalah untuk memberikan solusi pada rumusan masalah yaitu:

1. Menghasilkan tangki penampung air dari *output cooling tower* berdasarkan hukum kesetimbangan massa.
2. Menghasilkan sistem pengendalian level pada tangki penampung air agar sesuai *set point* dan dapat diamati pada *control room*.

1.4 Lingkup Tugas Akhir

Untuk memfokuskan penyelesaian masalah pada penelitian tugas akhir ini maka lingkup tugas akhir yang diangkat adalah sebagai berikut:

1. Tangki yang akan dipergunakan untuk menampung air dari *outlet cooling tower* akan dirancang dengan menggunakan Hukum Kesetimbangan Massa yang dalam hal ini adalah:

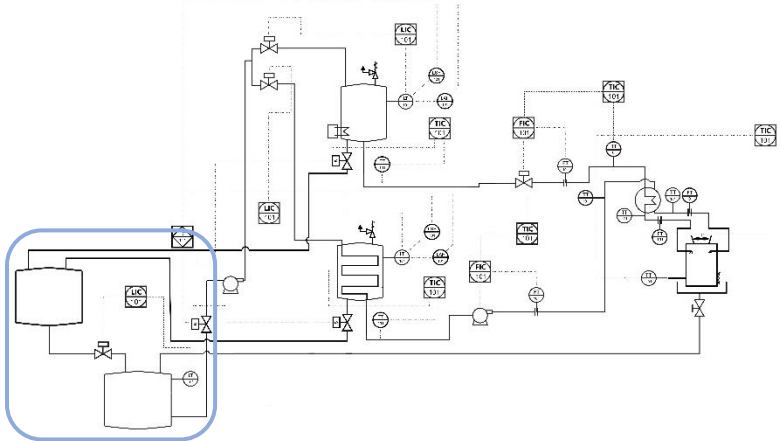
$$Q_{in1} + Q_{in2} = Q_{out} + \text{Akumulasi}$$
 dimana :
 Q_{in1} = Debit yang keluar dari *cooling tower*,
 Q_{in2} = Debit yang keluar dari *make up water tank*,
 Q_{out} = Debit yang dipompa pada tangki pemanas dan Pendingin,
 Akumulasi = Volume pada tangki penampung.
2. Akumulasi volume pada tangki penampung air akan diukur perubahan levelnya dengan menerima data yang dikirim oleh sensor level.

3. *Output* dari sensor level akan dibaca oleh *controller* yang digunakan untuk mengatur debit *input* dari *make up water tank* ke dalam tangki penampung air agar volume tangki tetap terjaga.

Halaman ini sengaja dikosongkan

BAB II DASAR TEORI

2.1 Simulator Penukar Panas



Gambar 2.1 P&ID Simulator Alat Penukar Panas



Gambar 2.2 BFD Simulator Alat Penukar Panas

Alat penukar panas adalah suatu peralatan di mana terjadi perpindahan panas dari suatu fluida yang suhunya lebih tinggi kepada fluida lain yang suhunya lebih rendah.[2] Pada pengerjaan tugas akhir ini akan membahas tentang simulator alat penukar panas. Air dengan suhu normal pada tangki penampungan dipompa menuju tangki pemanas dan tangki pendingin secara bersamaan.

Volume maksimal yang dibutuhkan oleh tangki pemanas adalah 12 Liter dengan suhu 70°C. Sedangkan pada tangki pendingin volume maksimal yang dibutuhkan adalah 40 liter dengan suhu 20°C. Kemudian air tersebut masuk ke *heat exchanger* agar mengalami pertukaran panas dengan suhu keluaran 35-40°C dengan mengatur *flowrate* air yang masuk. Air tersebut perlu dikondisikan agar menjadi suhu normal 25°C dengan *cooling tower*. Keluaran dari *cooling tower* akan dialirkan ke tangki penampungan untuk disirkulasikan kembali karena pada simulator penukar panas ini prosesnya terjadi secara kontinyu.

Apabila pada tangki penampungan pengukuran level tidak sesuai *set point* karena adanya *loss* selama proses, maka akan mengalirkan air dari *water makeup tank* agar level kembali ke *set point*.

2.2 Tangki Penampungan

Tangki penampungan menjadi bagian yang penting dalam suatu proses industri kimia karena tangki penampung tidak hanya menjadi tempat penampungan bagi produk dan bahan baku tetapi juga menjaga kelancaran ketersediaan produk dan bahan baku serta dapat menjaga produk atau bahan baku dari kontaminan yang dapat menurunkan kualitas dari produk atau bahan baku.[3]

Pada simulator penukar panas ini, tangki penampungan digunakan sebagai tempat penampung air yang akan dipompa menuju tangki pemanas dan tangki pendingin dengan volume yang sama. Air yang keluar dari tangki pemanas dan tangki pendingin dengan suhu tertentu akan menuju ke *heat exchanger* agar mengalami proses pertukaran panas. Seterlah itu air akan didinginkan *cooling tower* agar mencapai suhu ambient dan dialirkan kembali menuju tangki penampungan. Selama proses pertukaran panas pada plant ini rentan adanya penguapan air (*loss*). Selain itu, debit air yang dipompa menuju tangki pemanas dan tangki pendingin lebih besar dari pada air yang didapatkan dari *cooling tower*, sehingga selama proses berjalan untuk memenuhi ketersediaan air dibutuhkan *water make up* dari *make up water tank*.

Pada tangki penampungan dapat diterapkan hukum kesetimbangan massa yaitu “*massa tidak di ciptakan dan tidak dimusnahkan*”. Adapun penulisan rumus hukum kekekalan massa adalah sebagai berikut :

$$Inflow = Outflow + Accumulation \quad (2.1)$$

$$Q_{in1} + Q_{in2} = Q_{out} + \text{Akumulasi} \quad (2.2)$$

dimana :

Q_{in1} = debit yang keluar dari *cooling tower*,

Q_{in2} = debit yang keluar dari *make up water tank*,

Q_{out} = debit yang dipompa pada tangki pemanas dan pendingin.

Akumulasi = volume pada tangki penampung.

Sedangkan untuk mengetahui nilai Q dapat dicari dengan rumus :

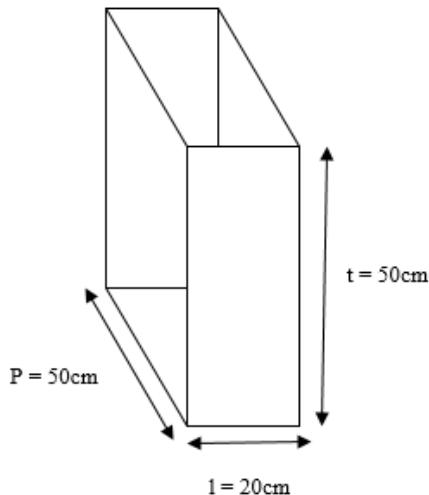
$$Q = V/t \quad (2.3)$$

dimana :

V = volume,

t = waktu.

Di bawah ini merupakan gambar tangki yang digunakan untuk tangki penampung pada simulator alat penukar panas.



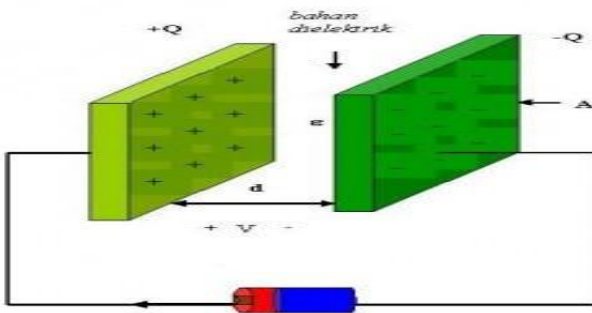
Gambar 2.3 Desain Tangki Penampungan

2.3 Sensor Kapasitansi

Kapasitor atau kapasitif adalah suatu komponen elektronika yang terdiri dari dua buah plat penghantar sejajar yang disekat satu sama lain dengan suatu bahan elektrik. Komponen – komponen ini sangat penting dalam elektronika karena memiliki sifat sifat seperti berikut :

1. Dapat menyimpan muatan listrik
2. Dapat menahan arus searah
3. Dapat melewatkan arus bolak – balik

Sensor kapasitif ini serupa dengan alat ukur regangan sebuah ukuran sel kapasitansi perubahan dalam karakteristik listrik. Kapasitif adalah sebuah alat yang menyimpan muatan listrik, terdiri dari pelat logam yang dipisahkan oleh isolator listrik, pelat logam yang terhubung ke sebuah sirkuit listrik eksternal. Muatan listrik dapat ditransfer dari satu pelat logam yang lain. Kapasitansi sebuah kapasitif yaitu ukuran kemampuannya untuk menyimpan muatan. Kapasitansi – kapasitansi dari suatu kapasitor berbanding lurus dengan daerah pelat logam dan berbanding terbalik dengan jarak antar kedua pelat logam. Ini juga tergantung pada karakteristik dari bahan isolasi yang berada diantara pelat dan kapasitor. Karakteristik ini disebut permitivitas adalah ukuran seberapa baik bahan isolasi meningkatkan kemampuan kapasitor menyimpan muatan.



Gambar 2.4 Konsep Dasar Kapasitansi

Satuan kapasitansi adalah coulomb / Volt (C/V) atau farad (F). Satu farad adalah jumlah muatan listrik sebesar satu coulomb yang disimpan didalam elektrik (zat perantara) dengan beda potensial sebesar satu volt. Kapasitansi suatu kapasitor bergantung pada :

1. Bahan dielektrik yang digunakan
2. Luas dari masing – masing pelat
3. Jarak antara kedua pelat

$$C = \epsilon \frac{A}{d} \dots\dots\dots(2.4)$$

dimana,

C = Kapasitansi dalam farad

A = Luas pelat

d = Jarak peat

ϵ = Permittivitas

Berdasarkan rumus diatas dapat diketahui bahwa sensor kapasitif merupakan sensor elektronika yang bekerja berdasarkan konsep kapasitif. Konsep kapasitor yang digunakan dalam sensor kapasitif adalah proses menyimpan dan melepas energi listrik dalam bentuk muatan – muatan listrik pada kapasitor yang dipengaruhi oleh luas permukaan, jarak dan bahan dielektrum.[4]

Tabel 2.1 Dielektrik *Constant* Bahan

Material	Dielektrik Constant (ϵ)
Udara kering (68°F)	1,000536
Air	4 - 88
Kayu kering	2 - 6
Kertas minyak	4
Kaca	3,7 - 10
Kertas	2,3

2.4 Stepper Motor Controlled Valve

Salah satu jenis *motor control valve* dc yang dikendalikan dengan pulsa-pulsa digital. Prinsip kerja motor stepper adalah bekerja dengan mengubah pulsa elektronis menjadi gerakan

mekanis diskrit dimana motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor stepper tersebut.



Gambar 2.6 *Stepper Motor Controlled Valve*

Stepping motor memiliki beberapa keunggulan - keunggulan sebagai berikut:

- a. Mudah dikendalikan dengan putaran sinkron yang hanya tergantung pada denyut yang diterima.
- b. Memiliki sudut yang tepat pada setiap langkah tanpa adanya akumulasi kesalahan sudut.
- c. Memiliki respon memulai dan berhenti yang baik pada saat digunakan, dan dapat digunakan sebagai servo motor.
- d. Putaran sinkron pada kecepatan yang sangat rendah dapat dicapai selagi “*carriage*” dihubungkan langsung pada shaft *stepping motor*.
- e. Memiliki ketahanan yang tinggi karena masa pengoperasiannya tergantung pada ketahanan “*bearing*” dan tidak pada bagian yang menempel semisal sikat atau “*brush*”.
- f. Berbagai macam sudut dapat dicapai tergantung dari tipe motor.

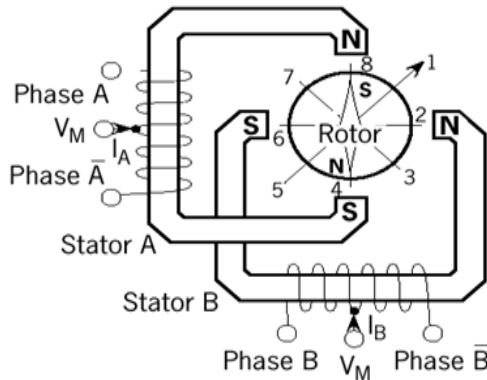
- g. Memiliki torsi menahan di posisi yang diinginkan pada saat motor dalam keadaan diam.
- h. Tidak terdapat sisa arus yang mengalir meskipun putaran terhalang oleh kelebihan beban.
- i. Kecepatan putaran tidak dipengaruhi oleh pergantian beban, karenanya kecepatan dapat selalu terjaga secara simultan dengan denyut masuk.

Motor stepper disini berfungsi sebagai penggerak valve secara elektrik. Motor stepper adalah perangkat elektromekanis yang bekerja dengan mengubah pulsa elektronis menjadi gerakan mekanis diskrit. Motor stepper bergerak berdasarkan urutan pulsa yang diberikan kepada motor. Karena itu, untuk menggerakkan motor stepper diperlukan pengendali motor stepper yang membangkitkan pulsa-pulsa periodik. Penggunaan motor stepper memiliki beberapa keunggulan dibandingkan dengan penggunaan motor DC biasa. Keunggulannya antara lain adalah :

- Sudut rotasi motor proporsional dengan pulsa masukan sehingga lebih mudah diatur.
- Motor dapat langsung memberikan torsi penuh pada saat mulai bergerak.
- Posisi dan pergerakan repetisinya dapat ditentukan secara presisi.
- Memiliki respon yang sangat baik terhadap mulai, stop dan berbalik (perputaran).
- Sangat realibel karena tidak adanya sikat yang bersentuhan dengan rotor seperti pada motor DC.
- Dapat menghasilkan perputaran yang lambat sehingga beban dapat dikopel langsung ke porosnya.
- Frekuensi perputaran dapat ditentukan secara bebas dan mudah pada *range* yang luas.

Berdasarkan metode perancangan rangkain pengendalinya, motor stepper dapat dibagi menjadi jenis unipolar dan bipolar. Rangkaian pengendali motor stepper unipolar lebih mudah dirancang karena hanya memerlukan satu switch / transistor setiap lilitannya. Untuk menjalankan dan menghentikan motor ini cukup dengan menerapkan pulsa digital yang hanya terdiri atas tegangan positif dan nol (*ground*) pada salah satu terminal lilitan (*wound*)

motor sementara terminal lainnya dicatu dengan tegangan positif konstan (V_M) pada bagian tengah (*center tap*) dari lilitan. [5]



Gambar 2.7 Motor Stepper dengan Lilitan Unipolar

2.5 Mikrokontroler AVR ATmega16

AVR merupakan seri Mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur *RISC (Reduced Instruction Set Computer)*. Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 *register general-purpose, timer/counter* fleksibel dengan mode *compare, interrupt* internal dan eksternal, serial *UART, programmable Watchdog Timer*, dan mode *power saving*. Beberapa diantaranya mempunyai ADC dan PWM internal. AVR juga mempunyai *In-System Programmable Flash on-chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. Chip AVR yang digunakan untuk tugas akhir ini adalah ATmega 16.

ATmega16 memiliki kapasitas EEPROM 1024 bytes dan kapasitas FLASH 16K words (32K bytes) sehingga program yang dibuat dapat memiliki ukuran besar. Gambar 2.28 menunjukkan mikrokontroler AVR ATmega 16.

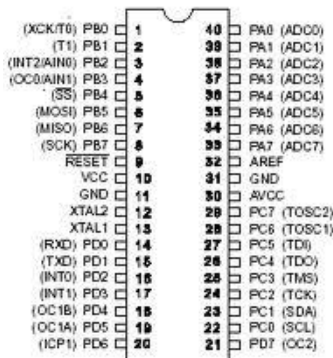


Gambar 2.8 Mikrokontroler AVR ATmega 16

2.5.1 Konfigurasi Pin ATmega16

Konfigurasi pin pada ATmega16 dengan kemasan 40-pin DIP (*Dual Inline Package*). Kemasan pin tersebut terdiri dari 4 *Port* yaitu *Port A*, *Port B*, *Port C* dan *Port D*. Masing masing *port* terdiri dari 8 buah pin. Selain itu juga terdapat pin RESET, VCC, GND 2 buah, VCC, XTAL1, XTAL2 dan AREF. [6]

Gambar berikut ini menunjukkan pin-pin pada mikrokontroler AVR ATmega 16.



Gambar 2.9 Pin-pin AVR ATmega 16

Deskripsi dari pin-pin ATmega16 dijelaskan sebagai berikut:

1. VCC merupakan pin yang berfungsi sebagai masukan catu daya
2. GND merupakan pin *ground*
3. PORT A (PA0,PA7) merupakan pin input/output dua arah dan pin masukan ADC.

4. PORT B (PB0-PB7) merupakan pin input/output dua arah dan pin yang memiliki fungsi khusus, seperti ditunjukkan oleh Tabel 2.2.

Tabel 2.2 Fungsi khusus port B

PIN	Fungsi Khusus
PB7	SCK (SPI Bus Serial Clock)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB5	MOSI (SPI Bus Master Output/Slave input)
PB4	SS (SPI Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OC0 (Timer/Counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB0	T0 T1 (Timer/Counter0 External Counter input) XCK (USART External Clock Input/Output)

5. PORT C (PC0-PC7) merupakan pin input/output dua arah dan pin yang memiliki fungsi khusus, seperti ditunjukkan oleh Tabel 2.3.

Tabel 2.3 Fungsi khusus port C

PIN	Fungsi Khusus
PC7	TOSC2 (Timer Oscillator pin2)
PC6	TOSC1 (Timer Oscillator pin1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two – Wire Serial Bus Data Input Output Line)
PC0	SCL (Two – Wire Serial Bus Clock Line)

6. PORT D (PD0-PD7) merupakan pin input/output dua arah dan pin yang memiliki fungsi khusus, seperti ditunjukkan oleh Tabel 2.4.

Tabel 2.4 Fungsi khusus port D

PIN	Fungsi Khusus
PD7	OC2 (Timer/Counter2 Output Compare Match Output)
PD6	ICP (Timer/Counter1 Input Capture Pin)
PD5	OC1A (Timer/Counter1 Output Compare A Match Output)
PD4	OC1B (Timer/Counter1 Output Compare B Match Output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

2.6 CodeVision AVR

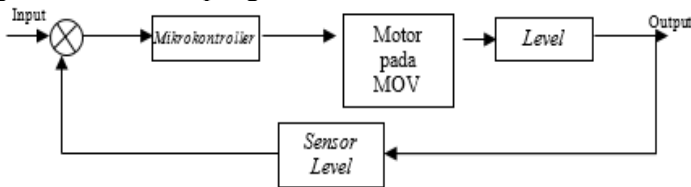
CodeVision AVR merupakan sebuah *software* yang digunakan untuk memprogram mikrokontroler sekarang ini telah umum. Mulai dari penggunaan untuk kontrol sederhana sampai kontrol yang cukup kompleks, mikrokontroler dapat berfungsi jika telah diisi sebuah program, pengisian program ini dapat dilakukan menggunakan compiler yang selanjutnya diprogram ke dalam mikrokontroler menggunakan fasilitas yang sudah di sediakan oleh program tersebut. Salah satu *compiler* program yang umum digunakan sekarang ini adalah CodeVision AVR yang menggunakan bahasa pemrograman C.

CodeVision AVR mempunyai suatu keunggulan dari compiler lain, yaitu adanya *codewizard*, fasilitas ini memudahkan pengguna atau pemrogram dalam inisialisasi mikrokontroler yang akan digunakan.[7]

2.7 Sistem Pengendalian Level

Sistem *control* atau sistem pengendalian adalah proses pengaturan / pengendalian terhadap satu atau beberapa besaran (variabel, parameter) sehingga berada pada suatu harga atau dalam suatu rangkuman harga (*range*) tertentu. Dalam istilah lain disebut juga teknik pengaturan, sistem pengendalian atau sistem pengontrolan. [8]

Salah satu penerapan sistem pengendalian adalah untuk menjaga ketinggian fluida pada suatu wadah. Berikut adalah diagram blok sistem pengendalian level.



Gambar 2.9 Diagram Blok Sistem Pengendalian Level

Sensor adalah elemen sistem yang secara efektif berhubungan dengan proses dimana suatu variabel sedang diukur dan menghasilkan suatu keluaran dalam bentuk tertentu tergantung pada variabel masukannya, dan dapat digunakan oleh bagian sistem pengukuran yang lain untuk mengenali nilai variabel tersebut.

Kontroler merupakan elemen sistem pengaturan yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*set point*) atau sinyal *error* mejadi sinyal kontrol. Sinyal *error* disini adalah selisih antara sinyal umpan balik yang dapat berupa sinyal keluaran plant sebenarnya atau sinyal keluaran terukur dengan sinyal masukan acuan (*set point*).

Aktuator adalah sebuah peralatan mekanis untuk menggerakkan atau mengontrol sebuah mekanisme atau sistem. Aktuator diaktifkan dengan menggunakan lengan mekanis yang biasanya digerakkan oleh motor listrik, yang dikendalikan oleh media pengontrol otomatis yang terprogram di antaranya mikrokontroler. Aktuator adalah elemen yang mengkonversikan besaran listrik analog menjadi besaran lainnya misalnya kecepatan putaran dan merupakan perangkat elektromagnetik yang menghasilkan daya gerakan sehingga dapat menghasilkan gerakan pada robot. Untuk meningkatkan tenaga mekanik aktuator ini dapat dipasang sistem *gearbox*. Aktuator dapat melakukan hal tertentu setelah mendapat perintah dari kontroller.

Plant adalah nama lain untuk sistem. Plant adalah seperangkat peralatan yang hanya terdiri dari beberapa bagian dan bekerja

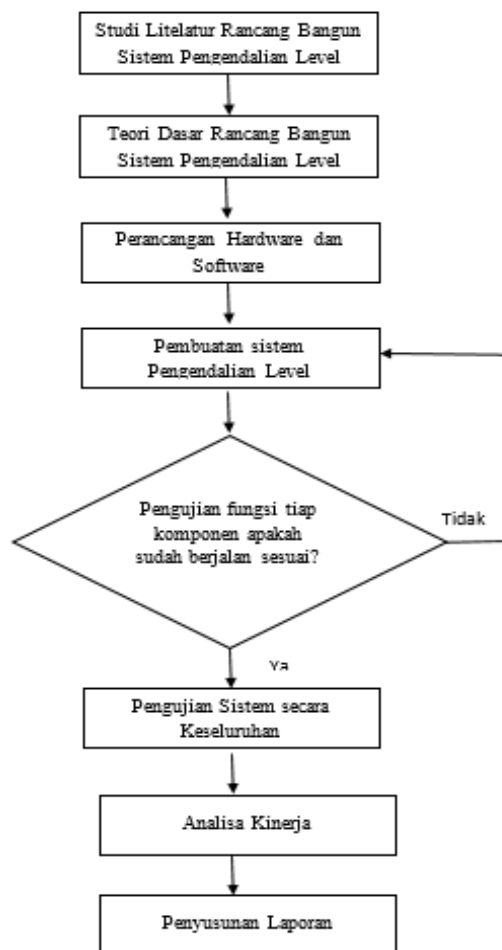
bersama-sama, yang digunakan untuk melakukan suatu operasi tertentu. Pada sistem pengendalian, setiap obyek fisik yang dikontrol disebut plant. [9]

Halaman ini sengaja dikosongkan

BAB III PERANCANGAN DAN PEMBUATAN ALAT

3.1. Diagram Alir (Flowchart)

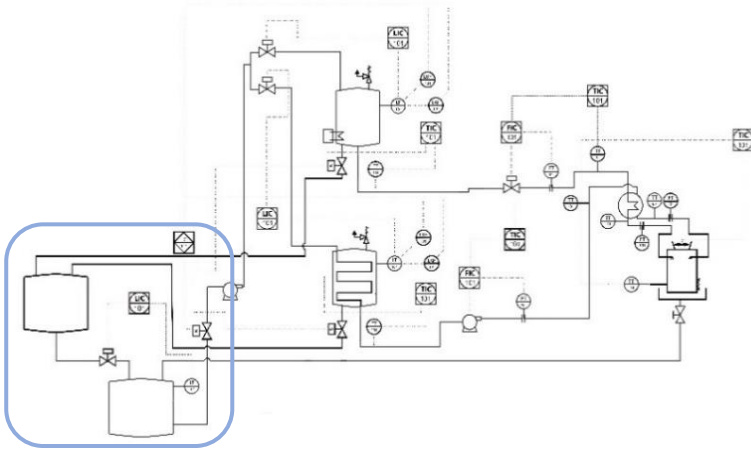
Langkah-langkah perancangan alat ini digambarkan dalam *flowchart* penelitian yang dapat dilihat pada gambar berikut ini :



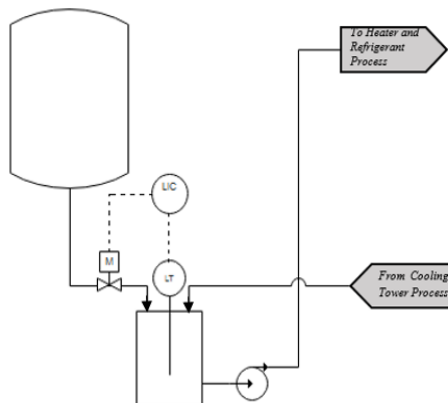
Gambar 3.1 Diagram Alir Pembuatan Alat

3.2 Perancangan Sistem

Perancangan sistem, untuk merancang baik dari *hardware* dan *software* yang digunakan.



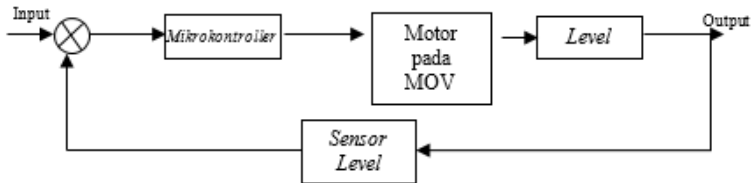
Gambar 3.2 P&ID Simulator Penukar Panas



Gambar 3.3 Perancangan Sistem Pengendalian Level pada Tangki Penampungan

Pada simulator alat penukar panas ini, tangki penampungan berfungsi sebagai tempat menampung fluida berupa

air keluaran dari *cooling tower* yang akan disirkulasi kembali. Karena selama proses memungkinkan adanya *loss*, untuk itu dibutuhkan air penambah yaitu *water makeup* untuk menjaga ketinggian air pada tangki penampung agar ketersediaan air saat sirkulasi tetap terjaga.

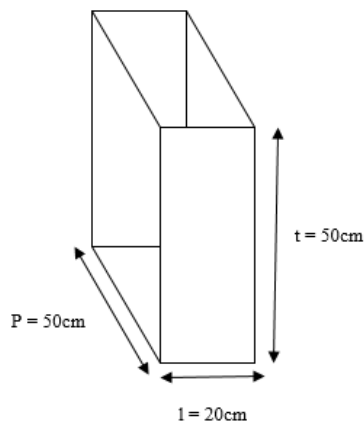


Gambar 3.4 Blok Diagram Sistem Pengendalian Level

Sistem pengendalian level pada tangki penampungan air dengan menggunakan sensor level yang akan mendeteksi ketinggian air dan mengirim sinyal ke *controller* yang akan mengirim sinyal untuk menggerakkan aktuator yang akan membuka aliran air pada pipa keluaran dari *make up water tank* jika level yang dideteksi tidak sesuai dengan *set point*.

3.2.1 Perancangan Hardware

Adapun perancangan hardware pada sistem pengendalian level adalah sebagai berikut:



Gambar 3.5 Desain Tangki Penampungan

Tangki penampungan yang digunakan adalah bentuk balok dan berbahan dasar akrilik dengan ketebalan 5mm yang mempunyai dimensi sebagai berikut :

$$Q_{in1} + Q_{in2} = Q_{out} + \text{akumulasi volume}$$

$$Q_{in \text{ total}} = Q_{out} + \frac{\text{akumulasi volume}}{t}$$

$$Q_{in \text{ total}} - Q_{out} = A \times \frac{dh}{dt}$$

$$\frac{Q_{in \text{ total}} - Q_{out}}{A} = \frac{dh}{dt}$$

Berdasarkan hukum kesetimbangan di atas, dapat diketahui ukuran tangki penampungan dengan menetapkan tinggi sebesar 50 cm, maka didapatkan ukuran tangki sebagai berikut:

$$V \text{ balok} = p \times l \times t$$

$$50.000 \text{ cm}^3 = p \times l \times 50 \text{ cm}$$

$$1.000 \text{ cm}^3 = p \times l$$

Sehingga :

$$p = 50 \text{ cm}, l = 20 \text{ cm}$$

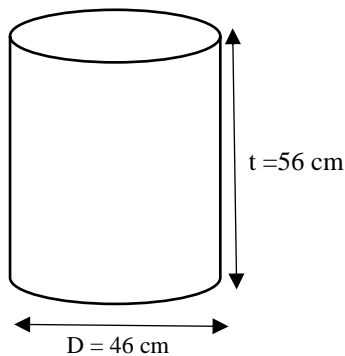
Pada tangki *water makeup* menggunakan bahan dasar plastik berbentuk tabung yang ditetapkan volume maksimalnya sebesar 100L. Ukuran dari tangki *water makeup* dapat diketahui dengan rumus volume tabung seperti di bawah ini:

$$\text{Volume tabung} = \pi r^2 t$$

$$= 3,14 \times 24 \text{ cm} \times 24 \text{ cm} \times 56 \text{ cm}$$

$$= 101.283 \text{ cm}^3$$

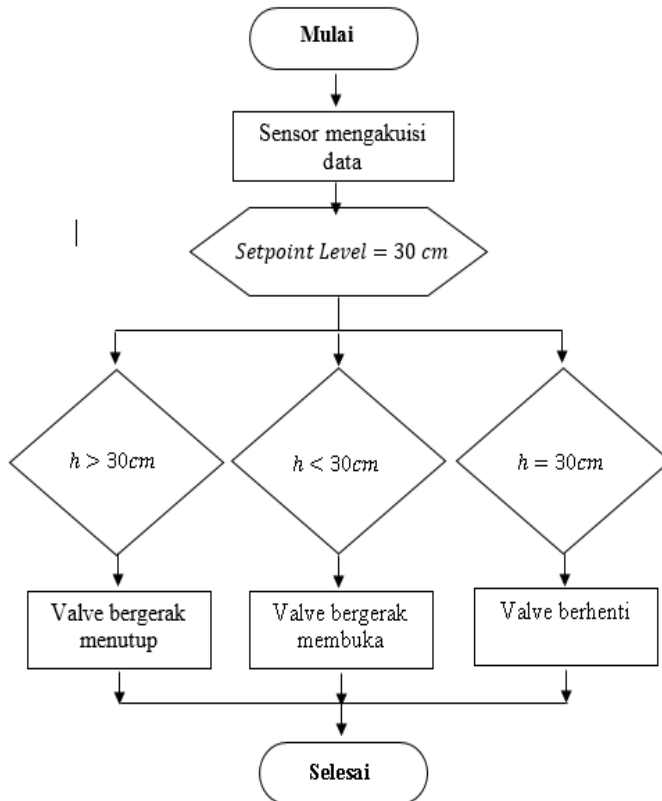
$$= 101,3 \text{ Liter}$$



Gambar 3.6 Desain Tangki *water makeup*

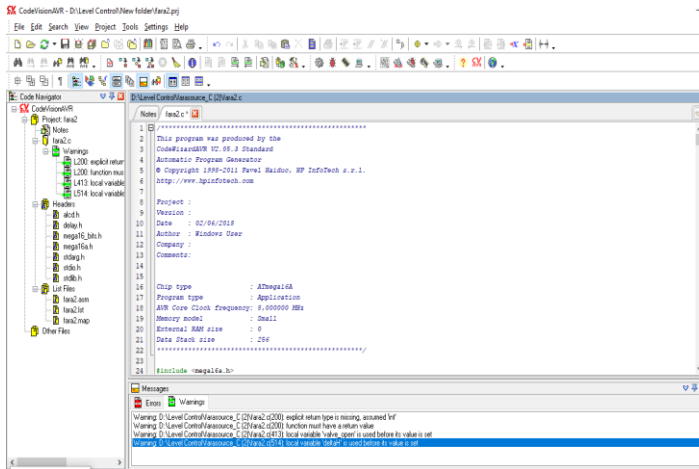
3.2.2 Perancangan Software

Adapun perancangan software pada sistem pengendalian level adalah sebagai berikut:



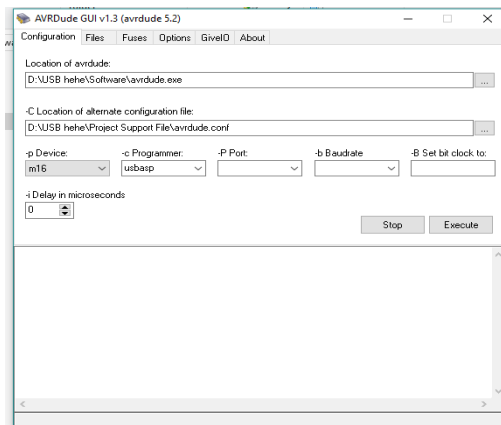
Gambar 3.7 Flowchart Program

Pada tahap ini perancangan *software* merupakan siklus gabungan untuk rangkaian pemrosesan sinyal. Rangkaian pemrosesan sinyal pada alat ini menggunakan mikrokontroler Atmega 16. Untuk membuat program pada Atmega 16, dibutuhkan *software* CodeVision AVR. Pada program ini, akan diberikan *listing program* untuk mengolah sinyal masukan. Sinyal masukan dari *sensing element*.



Gambar 3.8 Tampilan Listing Program *Software* CodeVision AVR

Software ini digunakan untuk meng-*compile* listing program yang dibuat pada *code vision* AVR ke mikrokontroler atmega 16. Selain itu, AVR Dunde 5.2 juga dapat melakukan pengaturan *clock* dan proses eksekusi program pada mikrokontroler.



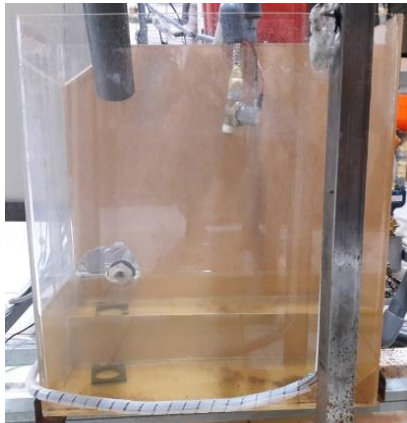
Gambar 3.9 Tampilan AVR Dunde 5.2

3.3 Pembuatan Sistem

Pada tahap pembuatan alat ini akan dibagi menjadi 2 bagian, yaitu pembuatan *hardware* alat dan pembuatan *software* alat. Untuk pembuatan *hardware* terdiri dari pembuatan elektik dan mekanik pada alat, sedangkan pembuatan *software* berisi langkah-langkah cara mengkoding mikrokontroler agar sistem *control* dapat bekerja sesuai dengan yang diinginkan.

3.3.1 Pembuatan *Hardware*

Pada pembuatan *hardware* ini meliputi pembuatan tangki. Tangki penampungan berbentuk balok dengan ukuran tinggi 50 cm, lebar 20 cm, dan panjang 50 cm. Tangki ini menggunakan bahan akrilik dengan tebal 5mm desain dari tangki penampungan.

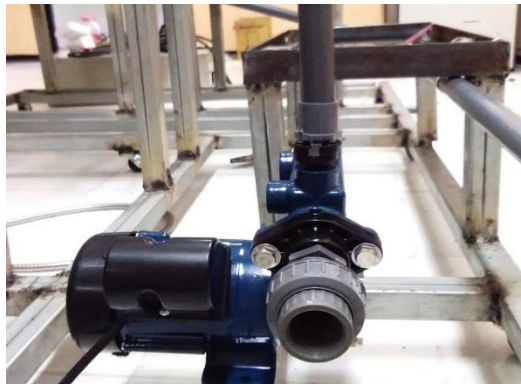


Gambar 3. 10 Tangki Penampungan

Adapun untuk *makeup water tank* menggunakan tangki berbahan plastik dengan ukuran maksimal 100 liter. Dengan pertimbangan untuk pengisian awal tangki pemanas dan pendingin membutuhkan kapasitas masing – masing sebesar 12 liter dan 40 liter, sedangkan sisanya digunakan sebagai sistem air penambah. Berikut adalah gambar tangki *make up water*.



Gambar 3.11 Tangki *makeup water*



Gambar 3.12 Pompa

Pompa yang digunakan adalah pompa sumur dangkal panasonic GP-75JXX dengan debit 16 liter/menit karena menyesuaikan kemampuan pada tangki pemanas yaitu 4 liter/menit dan tangki pendingin 8 liter/menit.

Berikut adalah spesifikasi pompa yang digunakan:

Voltase	: 220V/50Hz
Kapasitas	: 16 liter/menit
Ukuran Pipa Hisap	: 3/4"
Ukuran Pipa Dorong	: 3/4"

Daya Dorong : 11 m
Daya Hisap : 9 m
Total Head : 20 meter
Dimensi : 161 x 218 x 194 mm
Daya Listrik 75 Watt (*Output*) / 120 Watt (*Input*)



Gambar 3.13 *Motor Operating Valve*

Adapun valve yang digunakan adalah ball valve ukuran $\frac{3}{4}$ inch dengan penggerak motor stepper.

Motor stepper digunakan untuk memutar valve yang memiliki spesifikasi 1,8 deg/step. Untuk memutar valve dari posisi awal tertutup hingga terbuka penuh adalah 1900 step.

Halaman ini sengaja dikosongkan

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Rancang Bangun Alat

Berikut ini adalah gambar rancang bangun simulator alat penukar panas.



Gambar 4. 1 Plant Simuator Alat Penukar Panas

4.1.1 Pembuatan Sistem Pengendalian Level

Berikut ini adalah perancangan sistem pengendalian level pada tangki penampungan plant simulator alat penukar panas.



Gambar 4. 2 Sistem Pengendalian Level

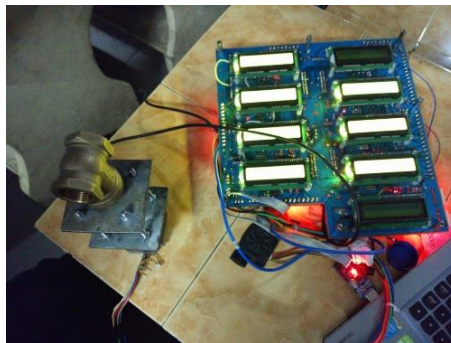
Pada simulator alat penukar panas ini, air keluaran dari *cooling tower* akan digunakan kembali untuk menyuplai kebutuhan air tangki pemanas dan pendingin untuk itu dibutuhkan tangki penampungan. Selama proses pasti terjadi *loss*, untuk itu level air pada tangki penampung harus dijaga agar supply tetap terpenuhi. Air yang digunakan sebagai air penambah berasal dari tangki *makeup water*. Proses pengendalian level ini menggunakan sensor level berbasis kapasitansi dengan aktuator berupa motor valve untuk mengatur debit yang masuk pada tangki penampung dengan *set point* level sebesar 30 cm.

4.1.2 Pengujian Komponen

Pengujian komponen dilakukan secara terpisah antara sensor level, motor valve, sensor flow.

4.1.2.1 Pengujian Motor Valve

Pengujian motor valve dilakukan untuk melihat daya yang dibutuhkan oleh valve dan waktu untuk bekerja dari posisi tertutup menjadi terbuka atau sebaliknya. Di bawah ini adalah gambar rangkaian motor valve yang digunakan pada sistem ini.



Gambar 4.3 Rangkaian *Motor Valve*

Pengujian motor valve ini dilakukan dengan menyambungkan *driver motor* dengan *power supply*. Untuk melihat daya yang dibutuhkan oleh motor untuk menggerakkan valve. Berikut adalah data pengujian motor stepper terhadap bukaan valve.

Tabel 4. 1 Hasil Pengujian Stepper

No	Valve buka LCD (%)	Valve buka (%)	Flow (LPM)
1	99	100	5,4
2	90	90	5,0
3	80	80	4,8
4	70	70	4,7
5	60	60	4,6
6	50	50	4,5
7	40	40	4,3
8	30	30	3,5
9	20	20	1,8
10	10	10	0,9
11	3	0	0

Dari data pengujian stepper di atas dapat diketahui besar bukaan valve berbanding lurus dengan besar aliran yang keluar dari *motor operated valve*. Apabila valve dalam keadaan buka 100 % maka, flow yang keluar sebesar 5,4 liter per menit.

4.1.2.2 Validasi Sensor Level

Sensor level yang digunakan, harus mampu membaca ketinggian pada tangki penampungan dengan sesuai, untuk itu dilihat dari karakteristik dari sensor level ini.

Pada pengujian sensor level ini dilakukan dengan menghubungkan sensor level yang ada pada tangki penampungan dengan mikrokontroler dan diberikan air sesuai dengan range pengukuran yang akan dibaca oleh sensor dan ditampilkan melalui LCD. Pada pengujian ini dilakukan dengan menetapkan 6 *range* pengukuran (15 cm, 18 cm, 21 cm, 24 cm, 27 cm, dan 30 cm) secara naik dan dilanjutkan dengan pengambilan data secara turun.

Berikut ini adalah data hasil pengujian naik sensor level yang dibandingkan dengan alat standar yaitu penggaris.

Tabel 4. 2 Pengambilan Data Naik

No	Pembacaan standar	Pembacaan Alat					Rata-Rata	Volt
		Data Ke-						
		1	2	3	4	5		
1	15	15	16	15	15	16	15,4	1,44
2	18	17	18	17	18	18	17,6	1,60
3	21	20	19	20	21	19	19,8	1,89
4	24	25	25	25	24	24	24,6	1,98
5	27	27	28	26	26	26	26,6	2,07
6	30	29	30	30	29	30	29,6	2,14

Berdasarkan data hasil pengujian naik di atas, dapat diketahui jika semakin besar ketinggian yang dideteksi sensor, maka semakin besar nilai tegangan yang dihasilkan. Selain itu, juga didapatkan data rata- rata pembacaan 6 *range* pengukuran yang telah ditetapkan.

Berikut ini adalah data hasil pengujian turun sensor level yang dibandingkan dengan alat standar yaitu penggaris.

Tabel 4. 3 Pengambilan Data Turun

No	Pembacaan standar	Pembacaan Alat					Rata-Rata	Volt
		Data Ke-						
		1	2	3	4	5		
1	15	16	16	16	16	15	15,8	1,44
2	18	17	18	18	17	18	17,6	1,51
3	21	21	20	21	20	19	20,2	1,56
4	24	24	24	24	25	25	24,4	1,57
5	27	26	26	26	28	27	26,6	1,78
6	30	30	29	30	29	30	29,6	1,82

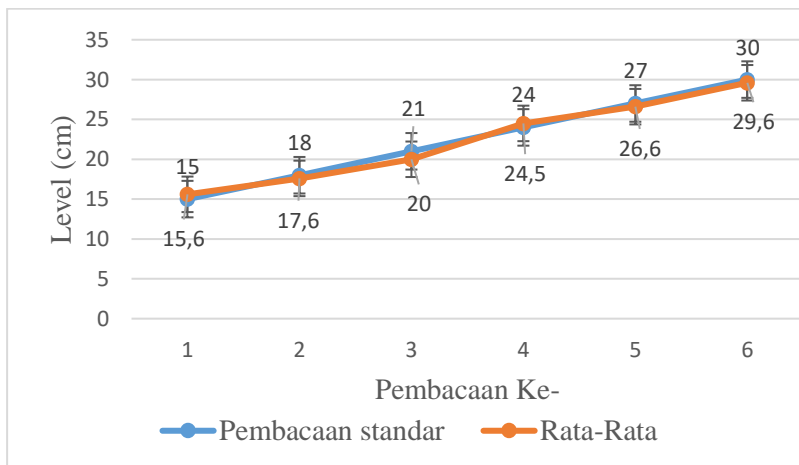
Berdasarkan data hasil pengujian turun di atas, dapat diketahui jika semakin besar ketinggian yang dideteksi sensor, maka semakin besar nilai tegangan yang dihasilkan. Selain itu, juga didapatkan data rata- rata pembacaan 6 *range* pengukuran yang telah ditetapkan.

Dari hasil pengujian data naik dan data turun diatas, maka diperoleh rata-rata setiap data naik dan turun untuk diolah lagi agar didapatkan nilai koreksi atau *error* antara data pembacaan alat ukur standar (penggaris) dengan pembacaan oleh sensor sebagai berikut:

Tabel 4.4 Rata – rata Pembacaan

No	Pembacaan standar	Pembacaan Alat		Rata-Rata	Volt
		Rata -rata Naik	Rata - rata Turun		
1	15	15,4	15,8	15,6	1,44
2	18	17,6	17,6	17,6	1,60
3	21	19,8	20,2	20	1,89
4	24	24,6	24,4	24,5	1,98
5	27	26,6	26,6	26,6	2,07
6	30	29,6	29,6	29,6	2,14

Dari tabel di atas didapatkan grafik perbandingan pembacaan rata-rata sensor dan alat ukur penggaris. Berikut ini adalah grafik perbandingan antara pembacaan oleh sensor dan penggaris.



Gambar 4.4 Grafik Pembacaan Sensor Level dan Penggaris

Berdasarkan grafik di atas dapat diketahui hasil pembacaan sensor level yang dibandingkan dengan alat standar yaitu penggaris. Pada pengujian sensor, diambil 6 titik yaitu 15 cm, 18 cm, 21 cm, 24 cm, 27 cm dan 30 cm dan dapat diketahui *error* pembacaan seperti pada grafik di atas.

4.2 Pengujian Sitem

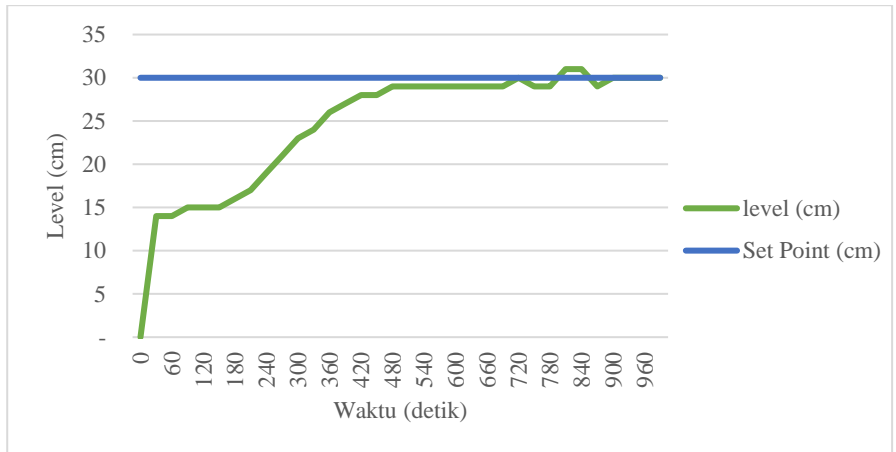
Pada pengambilan data didapatkan hasil grafik yang menggambarkan respon sistem untuk mencapai set point yaitu sebesar 30 cm. Pengujian dilakukan untuk mengetahui grafik respon *actuator* dan pembacaan level pada sensor berdasarkan waktu yang telah ditentukan. Adapun mode kontrol yang digunakan adalah Proporsional dengan nilai K_p sebesar 6.

Tabel 4.5 Hasil Pengujian Sistem

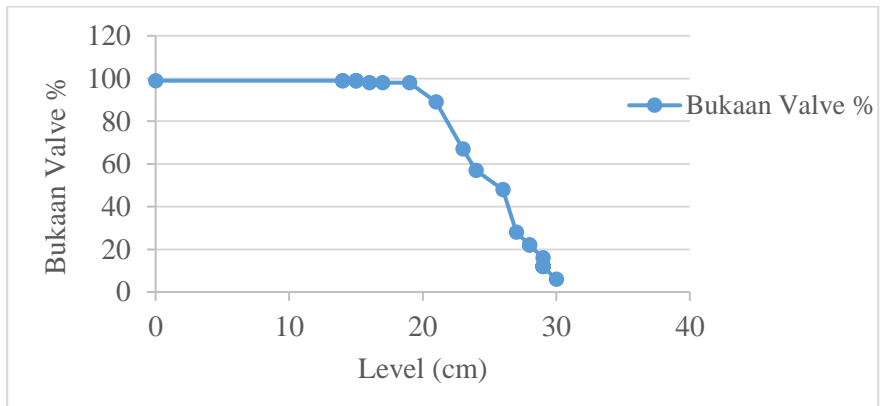
No	Waktu (detik)	Level (cm)	Bukaan Valve %
1	0	0	99
2	30	14	99
3	60	14	99
4	90	15	99
5	120	15	99
6	150	15	99
7	180	16	98
8	210	17	98
9	240	19	98
10	270	21	89
11	300	23	67
12	330	24	57
13	360	26	48
14	390	27	28
15	420	28	22
16	450	28	22

No	Waktu (detik)	Level (cm)	Bukaan Valve %
17	480	29	16
18	510	29	12
19	540	29	12
20	570	29	12
21	600	29	12
22	630	29	12
23	660	29	12
24	690	29	12
25	720	30	6
26	750	29	12
27	780	29	12
28	810	31	6
29	840	31	6
30	870	29	3
31	900	30	3
32	930	30	3
33	960	30	3
34	990	30	3

Dari tabel respon sistem di atas dapat dibuat grafik respon sistem dari *set point* level 30 cm dengan nilai K_p 6 adalah sebagai berikut:



Gambar 4.6 Grafik Respon Sistem pada *setpoint* 30 cm dengan Nilai K_p 6



Gambar 4.7 Grafik Respon Buka Valve Terhadap Level pada *set point* 30 cm

Berdasarkan grafik diatas dapat diketahui jika saat level kurang dari 15 maka valve akan membuka penuh dan semakin menutup dan mempertahankan nilai saat mendekati *set point* yaitu 30 cm. Pada grafik 4.6 dapat diketahui diketahui nilai *rise time* (T_r)

sebesar 720 detik, *peak time* pada 810 detik dan waktu *steady* sebesar 900 detik.

4.3 Pembahasan

Pada tugas akhir ini membahas tentang sistem pengendalian level tangki penampungan pada simulator alat penukar panas. Pada pengendalian level ini menggunakan sensor level berbasis kapasitansi. Telah dilakukan pengujian sensor level dengan dua cara yaitu pengambilan data naik dan pengambilan data turun. Adapun variasi pengukuran yaitu 15 cm, 18 cm, 21 cm, 24 cm, 27 cm, 30 cm dengan 5 kali pengambilan data. Dari kedua data tersebut akan dirata – rata dan hasil pembacaanya akan dibandingkan dengan alat standar yaitu penggaris yang hasilnya bisa dilihat pada gambar 4.4 grafik pembacaan sensor dengan penggaris. Untuk pengujian motor valve dengan mengatur % bukaan valve lalu dibandingkan hasilnya pada tampilan LCD dan dilihat nilai *flow* yang mengalir.

Pada pengendalian level ini menggunakan mode kontrol proporsional dengan metode *trial and error*. Adapun *set point* level adalah 30 cm dengan menggunakan nilai *kp* sebesar 6. Berdasarkan data hasil pengujian sistem dengan nilai *set point* 30cm dapat diketahui nilai *rise time* (*Tr*) sebesar 720 detik, *peak time* pada 810 detik dan waktu *steady* sebesar 900 detik seperti pada gambar 4.6 grafik respon sistem pada *setpoint* 30 cm dengan nilai *kp* 6.

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian laju aliran pada perancangan sistem pengendalian level tangki penampug pada simulator alat penukar panas ini, dapat disimpulkan, yaitu:

1. Didapatkan ukuran tangki penampungan cooling tower dengan ukuran sebesar 50x20x50 cm.
2. Berdasarkan data hasil uji respon sistem didapatkan nilai *rise time* (T_r) sebesar 720 detik, *peak time* atau maksimum *overshoot* pada 810 detik dan waktu *steady* sebesar 900 detik.

5.2 Saran

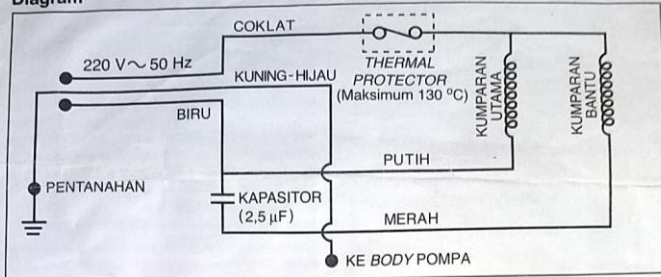
Adapun saran untuk penelitian selanjutnya agar hasil yang dicapai dapat memenuhi harapan, antara lain:

1. Lebih diperhatikan dalam penentuan pemilihan sensor dan aktuator agar mudah untuk disinkronkan.
2. Untuk menghindari *loss* saat pengendalian, dipastikan bahwa plant sudah tidak ada yang bocor karena akan mempengaruhi pengukuran *flow*.

Halaman ini sengaja dikosongkan

LAMPIRAN A MANUAL BOOK POMPA

Diagram



Spesifikasi Teknik

MODEL	GL-75JXK
Motor	Induksi / 1 fasa
Sumber tegangan	220 V ~ 50 Hz
Daya keluaran	75 Watt
Arus masukan	0,7 Ampere
Jumlah kutub	2
Daya hisap maksimum (Hs Maks.)	9 meter
Daya dorong maksimum (Hd Maks.)	8 meter
Tinggi aliran maksimum (Ht Maks.)	17 meter
Kapasitas air maksimum	16 l/menit
Pipa hisap / dorong	3/4 inci
Ukuran	217,5 x 160,5 x 194 mm
Berat bersih / kotor	3,7 kg / 4,2 kg

Catatan : Desain dan Spesifikasi Teknik dapat berubah tanpa pemberitahuan sebelumnya.

Produsen :

PT Panasonic Manufacturing Indonesia

Jl. Raya Bogor Km. 29, Pekayon, Pasar Rebo, Jakarta 13710, Indonesia
Tel.: 62-21-8710221/Fax.: 62-21-8710851 <http://www.panasonic.co.id>

Sales & Servis :

PT Panasonic Gobel Indonesia

Kantor Pusat :

Jl. Dewi Sartika (Cawang II), Jakarta 13630 Telp. (021)-8090108, 8015710 Fax. (021)-8004368

Pusat Pelayanan Pelanggan (PPP) :

Jl. Dewi Sartika (Cawang II), Jakarta 13630 Telp. (021)-8009494 Fax. (021)-8011118



© PT Panasonic Manufacturing Indonesia 2014

P0314H0

ZGP75JXDA

LAMPIRAN B
LISTING PROGRAM CV AVR

/******

This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :
Version :
Date : 02/06/2018
Author : Windows User
Company :
Comments:

Chip type : ATmega16A
Program type : Application
AVR Core Clock frequency: 8,000000 MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256

*****/

```
#include <mega16a.h>
#include <stdio.h>
#include <delay.h>
#include <stdlib.h>
// Alphanumeric LCD functions
#include <alcd.h>
#include <alcd.h>
#define stepB PORTB.0
#define stepD PORTB.1
#define stepC PORTB.3
#define stepA PORTB.4
#define ledrefill PORTD.5
#define ledrun PORTD.6
```

```

#define pump PORTD.7
#define maxstep 1900
#define backlight PORTC.7

#define Kp 6
#define Kd 0
int detik,menit,minperiode,refill,periode,respon,direct,stepcount;
unsigned int pulse;

bit refresh,play,override, stepper;

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
//flow1++; //flow cooling tower
periode=pulse;
minperiode=TCNT2;
TCNT2=0;
pulse=0;

}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here
//flow water makeup
refill=0;
menit=0;
}

// External Interrupt 2 service routine
interrupt [EXT_INT2] void ext_int2_isr(void)
{
// Place your code here
play=~play; //inverse kondisi button

```

```
}
```

```
#ifndef RXB8  
#define RXB8 1  
#endif
```

```
#ifndef TXB8  
#define TXB8 0  
#endif
```

```
#ifndef UPE  
#define UPE 2  
#endif
```

```
#ifndef DOR  
#define DOR 3  
#endif
```

```
#ifndef FE  
#define FE 4  
#endif
```

```
#ifndef UDRE  
#define UDRE 5  
#endif
```

```
#ifndef RXC  
#define RXC 7  
#endif
```

```
#define FRAMING_ERROR (1<<FE)  
#define PARITY_ERROR (1<<UPE)  
#define DATA_OVERRUN (1<<DOR)  
#define DATA_REGISTER_EMPTY (1<<UDRE)  
#define RX_COMPLETE (1<<RXC)
```

```
// USART Receiver buffer  
#define RX_BUFFER_SIZE 6
```



```

char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if(data==0x23)           // Inisial stx (0x23 = #)
{
rx_counter=0;           // rx_counter mulai dari 0
rx_wr_index=0;          // rx_wr_index mulai dari 0
}
if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
{
rx_buffer[rx_wr_index++]=data;
#if RX_BUFFER_SIZE == 256
// special case for receiver buffer size=256
if (++rx_counter == 0) rx_buffer_overflow=1;
#else
if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if (++rx_counter == RX_BUFFER_SIZE)
{
rx_counter=0;
rx_buffer_overflow=1;
}
}
#endif
}

```

```
}  
}
```

```
#ifndef _DEBUG_TERMINAL_IO_  
// Get a character from the USART Receiver buffer  
#define _ALTERNATE_GETCHAR_  
#pragma used+  
char getchar(void)  
{  
    char data;  
    while (rx_counter==0);  
    data=rx_buffer[rx_rd_index++];  
    #if RX_BUFFER_SIZE != 256  
    if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;  
    #endif  
    #asm("cli")  
    --rx_counter;  
    #asm("sei")  
    return data;  
}  
#pragma used-  
#endif
```

```
// Standard Input/Output functions  
#include <stdio.h>
```

```
// Timer 0 overflow interrupt service routine  
interrupt [TIM0_OVF] void timer0_ovf_isr(void)  
{  
    // Place your code here  
    respon++;  
  
    if (respon>=61)  
    {  
        refresh=1;  
        respon=0;  
    }  
}
```

```

// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Place your code here
stepper=1;
}

// Timer2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
// Place your code here
pulse++;
}

// Declare your global variables here
moving_stepper(int i)
{
    if (direct==1) //valve buka
    { switch (i)
      {
        case 0: {stepA=0; stepB=0; stepC=0; stepD=0; break;}
        case 1: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
        case 2: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
        case 3: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 4: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
        case 5: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 6: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
        case 7: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
        case 8: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
      } stepcount++;if
(stepcount>=maxstep){ direct=0;stepcount=maxstep;stepA=0;
stepB=0; stepC=0; stepD=0;}
    }
    else if(direct==2) //valve tutup
    {
switch (i)

```

```

{
    case 0: {stepA=0; stepB=0; stepC=0; stepD=0; break;}
    case 1: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
    case 2: {stepA=1; stepB=0; stepC=0; stepD=0; break;}
    case 3: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
    case 4: {stepA=0; stepB=0; stepC=1; stepD=0; break;}
    case 5: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
    case 6: {stepA=0; stepB=1; stepC=0; stepD=0; break;}
    case 7: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
    case 8: {stepA=0; stepB=0; stepC=0; stepD=1; break;}
} stepcount--
;if(stepcount<=0){stepcount=0;direct=0;stepA=0; stepB=0;
stepC=0; stepD=0;}
}
    else if (direct==0){stepA=0; stepB=0; stepC=0; stepD=0;}
//valve mati

}

void main(void)
{
// Declare your local variables here
// int stepA,stepB,stepC,stepD;
    int tinggi, arus, valve_open, frek,
target_valve, p, lpm, remote_valve, derivative, deltaH, awalerror;
    unsigned char temp[16]; unsigned char skip[3];
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization

```

```

// Func7=In Func6=In Func5=In Func4=Out Func3=In
Func2=Out Func1=Out Func0=Out
// State7=T State6=T State5=T State4=0 State3=P State2=0
State1=0 State0=0
PORTB=0x04;
DDRB=0x1B;

// Port C initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out
Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=In
Func2=In Func1=Out Func0=In
// State7=0 State6=0 State5=0 State4=0 State3=P State2=P
State1=0 State0=T
PORTD=0x1C;
DDRD=0xE2;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 31,250 kHz
// Mode: Normal top=0xFF
// OC0 output: Disconnected
TCCR0=0x04;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 125,000 kHz
// Mode: Ph. & fr. cor. PWM top=OCR1A
// OC1A output: Discon.

```

```
// OC1B output: Discon.  
// Noise Canceler: Off  
// Input Capture on Falling Edge  
// Timer1 Overflow Interrupt: On  
// Input Capture Interrupt: Off  
// Compare A Match Interrupt: Off  
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x03;  
TCCR1B=0x1C;  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x01;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;
```

```
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: 250,000 kHz  
// Mode: Normal top=0xFF  
// OC2 output: Disconnected
```

```
ASSR=0x00;  
TCCR2=0x02;  
TCNT2=0x00;  
OCR2=0x00;
```

```
// External Interrupt(s) initialization  
// INT0: On  
// INT0 Mode: Falling Edge  
// INT1: On  
// INT1 Mode: Falling Edge  
// INT2: On  
// INT2 Mode: Falling Edge  
GICR|=0xE0;
```

```
MCUCR=0x0A;
MCUCSR=0x00;
GIFR=0xE0;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x45;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;
```

```

// Alphanumeric LCD initialization
// Connections are specified in the
// Project\Configure\C Compiler\Libraries\Alphanumeric LCD
menu:
// RS - PORTC Bit 0
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 3
// D5 - PORTC Bit 4
// D6 - PORTC Bit 5
// D7 - PORTC Bit 6
// Characters/line: 8
lcd_init(16);
// Global enable interrupts
#asm("sei")
lcd_init(16);
lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Level Control");
lcd_gotoxy(0,1);
lcd_putsf("Faradhiba AS");
backlight=1;
tinggi=0;
stepcount=maxstep;direct=2;//tutup valve
p=0;
while(stepcount>0)
{ if (stepper==1)
    { p++;
      if (p>=9) p=1;
      if(PINA.7==0)
      {
        arus=0;
        arus=arus+(1*PINA.0);
        arus=arus+(2*PINA.1);
        arus=arus+(4*PINA.2);
        arus=arus+(8*PINA.3);
        arus=arus+(16*PINA.4);
        arus=arus+(32*PINA.5);
      }
    }
}

```



```

        arus=arus+(64*PINA.6);
    }

    moving_stepper(p);stepper=0;
}
};
direct=0;
lcd_clear();
target_valve=0;
override=0;
play=1;
while (1)
    {
        if(rx_buffer_overflow==1)    //penanda flag awal data
        {
            if ((rx_buffer[0]==0x23)&&(rx_buffer[5]==0x25))
//jika rx_buffer char ke [0]=# dan char ke [5]=%
            {
                switch(rx_buffer[1])    //opcode data ke
                {
                    case 49:{ printf ("01;");break;} //opcode buat
ngeread
                    case 50:{ printf("%d;",tinggi);break;}    //opcode
buat ngeread data level
                    case 51:{ printf("%d;",refill);break;}    //opcode
buat ngeread total air yg digunakan
                    case 52:{ printf("%d;",valve_open);break;}
//opcode buat ngeread persen bukaan valve dari water make up
                    case 53:{ printf ("%d;",play);break;}    // read
status aktif plan

                    case 54:{ if(pump==1) printf("Pompa Menyala;\n");
                        else if(pump==0) printf("Pompa Mati;\n");
                        break;}
                    case 55:{ skip[0]=rx_buffer[2];
                        skip[1]=rx_buffer[3];
                        skip[2]=rx_buffer[4];
                        remote_valve=atoi(skip);

```

```

        break;} //opcode buat
ngatur bukaan valve
        case 56:{ if(rx_buffer[4]=='1')pump=1;
                 else if(rx_buffer[4]=='0')pump=0;
                 break;} //opcode buat
on off pump
        case 126:{ if(rx_buffer[4]=='1')play=1;
                  else if(rx_buffer[4]=='0')play=0;
                  break;} //opcode buat
on off loop lcu
        case 127:{ if(rx_buffer[4]=='1')backlight=1;
                  else if(rx_buffer[4]=='0')backlight=0;
                  break;}
        case 128:{ if(rx_buffer[4]=='1')override=1;
//taking over by hmi
                  else if(rx_buffer[4]=='0')override=0;
                  break;}
        }
    }
else
    { printf ("error");}
  rx_buffer_overflow=0;
}

if (stepper==1)
{ p++;
  if (p>=9) p=1;
  stepper=0;
  if(PINA.7==1)
  {
    tinggi=0;
    tinggi=tinggi+(1*PINA.0);
    tinggi=tinggi+(2*PINA.1);
    tinggi=tinggi+(4*PINA.2);
    tinggi=tinggi+(8*PINA.3);
    tinggi=tinggi+(16*PINA.4);
    tinggi=tinggi+(32*PINA.5);
    tinggi=tinggi+(64*PINA.6);
  }
}

```

```

    }

    moving_stepper(p);
    stepper=0;
}

// Place your code here
if (refresh==1)
{

    lcd_gotoxy(0,0);
    lcd_putsf("h= ");
    sprintf(temp,"%2dcm",tinggi);
    lcd_puts(temp);

    if((periode==0)&&(minperiode==0))frek=0; //flow

    frek=10000/((periode*0.256)+(minperiode*0.001));
    if(pulse>=613) {frek=0;}
    lpm=frek*0.127292;
    //refill=refill+(lpm/2); //0.5s=f
    refill=lpm;
    lcd_gotoxy(8,0);
    lcd_putsf("Rf="); //tampilan lcd
    sprintf(temp,"%4d",refill);
    lcd_puts(temp);

    valve_open=stepcount/(maxstep/100);
    lcd_gotoxy(0,1);
    lcd_putsf("val=");
    sprintf(temp,"%3d",valve_open);
    lcd_puts(temp);

    detik++; //waktu
    if (detik >=120){menit++;detik=0;}
    lcd_gotoxy(9,1);
    sprintf(temp,"%4dm",menit);

```

```

lcd_puts(temp);

if(PIND.4==1)ledrefill=1;
else ledrefill=0;

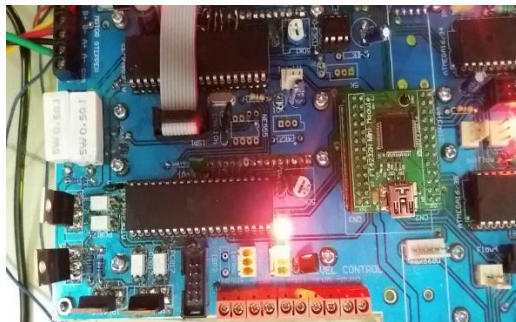
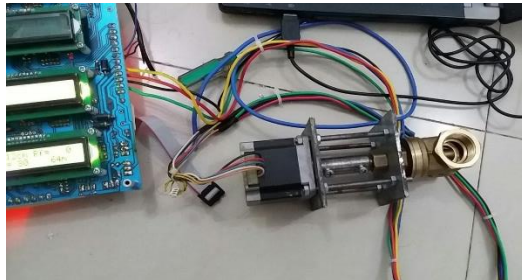
if (play==1) {
    if(tinggi>=15)pump=1;
        else pump=0;
        ledrun=1;
        //progam penegndalian motor stepper
    if (override==1)target_valve=remote_valve;
    if (target_valve>=valve_open+5)direct=1;
    else if (target_valve<=valve_open-5)direct=2;
    else if
        ((target_valve<=valve_open+4)&&(target_valve>=valve_
        open-4))direct=0;

    //progam setpoint valve
    awalerror=deltaH;
    deltaH= 30-tinggi;
    derivative= deltaH-awalerror;
    target_valve=(Kp * deltaH)+(Kd* derivative);
    if (target_valve>=99) target_valve=100;
    else if(target_valve<=1)target_valve=0;
        }

else
    {pump=0; if (stepcount>=1)direct=2;
    else direct=0;
    menit=0;refill=0;
    ledrun=0;}
refresh=0;
}
}
}

```

LAMPIRAN C DOKUMENTASI



BIODATA PENULIS



Faradhiba Alifiyah Safitri yang biasa dipanggil “Fara atau dhiba” ini merupakan mahasiswi dari daerah Sidoarjo Jawa Timur yang dilahirkan pada tanggal 27 Oktober 1997. Penulis menempuh pendidikan formal di MI Islamiyah Cangkringsari, SMP Negeri 2 Sukodono, Dan SMAN 1 Taman. Kemudian pada tahun 2015 penulis mengikuti tes DIII ITS dan diterima pada jurusan DIII Teknik Instrumentasi dan terdaftar dengan NRP 105115000000003.

Pada tahun 2018 ini, penulis mampu menyelesaikan gelar ahli madya Program Studi DIII Teknik Instrumentasi, Departemen Teknik Instrumentasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya. Penulis berhasil menyelesaikan Tugas Akhir dengan judul **“RANCANG BANGUN SISTEM PENGENDALIAN TANGKI PENAMPUNGAN PADA SIMULATOR ALAT PENUKAR PANAS”**. Apabila ada pertanyaan mengenai tugas akhir penulis dapat menghubungi 085655098304 atau email : faradhib@gmail.com